# Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs[*]

Xin Zhao, Dean L. Lewis, Hsien-Hsin S. Lee and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.
{xinzhao, dean, leehs, limsk}@ece.gatech.edu

## ABSTRACT

Pre-bond testing of 3D stacked ICs involves testing individual dies before bonding. The overall yield of 3D ICs improves with pre-bond testability because designers can avoid stacking defective dies with good ones. However, pre-bond testability presents unique challenges to 3D clock tree design. First, each die needs a complete 2D clock tree for the pre-bond testing. In addition, the entire 3D stack needs a complete 3D clock tree for post-bond testing and normal operations. In the case of two-die stack, a straightforward solution is to have two complete 2D clock trees connected with a single Through-Silicon-Via (TSV). We show that this solution suffers from long wirelength and high clock power consumption. Instead, our algorithm minimizes the overall wirelength and clock power consumption while providing the pre-bond testability and post-bond operability under given skew and slew constraints. Compared with the single-TSV solution, SPICE simulation results show that our multi-TSV approach significantly reduces the clock power by up to 15.9% for two-die and 29.7% for four-die stack. In addition, the wirelength reduction is up to 24.4% and 42.0%.

## 1. INTRODUCTION

3D system integration has emerged as a key enabling technology to continue the scaling trajectory predicted by Moore's Law for future IC generations. Using 3D integration, the average and maximum communication distance between components placed on different dies can be substantially reduced, which further translates into significant savings on delay, power and area. Moreover, it enables the integration of heterogeneous devices, making the entire system compact and efficient. Nevertheless, the success of 3D stacked ICs is predicated on the final post-bond yield, i.e., to minimize the chances of bonding good dies with defective ones together. In other words, pre-bond testability must be provided prior to the bonding process to test each individual die, which may consist of partial functionality.

To tackle the testing issues for 3D stacked ICs, several testing methods were investigated. In [13], Wu *et al.* proposed 3D scan chain design approaches to improve testability. The stitching wirelength is minimized in their work. Lewis and Lee presented an architectural solution in [8] to the pre-bond testability for 3D die-stacked microprocessors. They discussed how to perform testing for functional modules that are splitted into multiple dies. They also investigated new design methods in [9] to address similar testing issues caused by partially functional pre-bond circuits.

Minz *et al.* [10] presented a 3D clock routing algorithm under the wirelength minimization goal. Their 3D tree has a unique property, where only one of the dies in the stack contains a fully connected 2D clock tree while the other dies contain many small, isolated subtrees. While this algorithm takes the advantage of TSVs to shorten the total wirelength of the clock signal, it makes pre-bond testing very difficult as a large number of test probes will be required to provide synchronous clock signals through these TSVs for testing those dies without a fully connected 2D clock tree. Their technique shows that multiple TSVs help reduce wirelength and clock power but complicate the pre-bond testing. Our work aims at addressing these issues and providing methods to design a pre-bond testable clock tree for 3D stacked ICs.

The contributions of our work are as follows: (1) We present the first work on pre-bond testable clock routing. We propose a new circuit element called *TSV-buffer*, which supports zero-skew pre-bond testing for the clock trees that use multi-TSVs. We also introduce so called *redundant tree*, which supports the pre-bond testing of dies that do not contain a fully connected clock tree. We show that these circuit elements are essential in supporting efficient pre-bond testing while minimizing the overall wirelength and clock power. (2) In order to improve the reliability of our pre-bond testable 3D clock tree, we develop a slew-aware merging and buffering method to keep the slew rate at clock sinks under the given constraint. This method also helps reduce wirelength and power consumption of the pre-bond testable 3D clock tree. (3) Compared with a straightforward solution, which uses a single TSV in between two dies for pre-bond testability, our solution reduces the wirelength and clock power consumption by up to 24.4% and 15.9% for two-die, and 42.0% and 29.7% for four-die stack.

## 2. TESTABLE CLOCK ROUTING

The pre-bond testable 3D clock routing problem is defined as follows: Given a set of sinks distributed on N dies $(N > 1)$ and an upper bound of TSV count, the goal is to construct a 3D clock tree such that during post-bond operations, the tree connects all the sinks and provides the clock signal with minimum skew;[1] and

[*]This material is based upon the work supported by the National Science Foundation under CAREER Grant No. CCF-0546382, Grant No. CCF-0811738, the Center for Circuit and System Solutions (C2S2), and the Interconnect Focus Center (IFC).

*ICCAD'09,* November 02–05, 2009, San Jose, California, USA.
Copyright 2009 ACM 978-1-60558-800-1/09/11 ...$10.00.

[1] Our zero-skew clock trees are built under the Elmore delay model. But, we report the clock skew based on SPICE simulation in the experiment section.

during pre-bond testing, a 2D clock tree, together with one test probe for each die, provides the clock signal to the sinks on the die with minimum skew. The objective is to minimize the wirelength and clock power under the given TSV budget and clock slew constraints.

## 2.1 Overview

We first develop a pre-bond testable clock routing algorithm for two-die stack. We then extend it to handle more than two dies in Section 2.6. The input to our algorithm includes the location and capacitance of sinks on both dies (= die-0 and die-1), and an upper bound of TSV usage ($> 1$). Assume that die-0 contains the clock source. Our algorithm consists of the following two main steps: (1) **3D tree construction**: the goal is to generate a 3D clock tree connecting all the sinks on both dies so that (a) the overall 3D tree has zero skew under the Elmore delay model, (b) total wirelength is minimized, and (c) die-0 contains a fully connected 2D tree with zero skew. In this case, the 3D tree is used during post-bond testing and normal operations, and the 2D tree on die-0 is used for pre-bond testing of die-0. We utilize so called "TSV-buffer" to make sure that the 2D tree on die-0 maintains zero skew during *both* pre-bond and post-bond. (2) **Redundant tree routing**: if multiple TSVs are used, the 3D tree construction step generates a 3D tree, where die-1 contains several sub-trees that are not connected. The goal of the redundant tree routing step is to connect the roots of the sub-trees on die-1 and form a single fully connected 2D tree so that (a) the skew is zero, and (b) the total wirelength is minimized. This 2D tree is used for the pre-bond testing of die-1. In addition, the additional tree used to connect the roots, so called "redundant tree", is disconnected during the post-bond operations. We use transmission gates (= TGs) to connect and disconnect this redundant tree.

## 2.2 Review of Existing Work

We use the 3D-MMM algorithm [10] to generate the abstract tree for the 3D clock sinks in a top-down manner. The basic idea is to recursively divide the given sink set into two subsets until each sink belongs to its own set. We then visit each sink in a bottom-up fashion and start merging subtrees until all sinks are connected via a single tree. At each recursive partitioning step, we divide the given sink set into two subsets $A$ and $B$. The following two cases are considered based on the TSV bound of the current sink set: (1) If the TSV bound is one, the current sink set is partitioned such that the sinks on the same die belong to the same subset. The connection between $A$ and $B$ needs one TSV. (2) If the TSV bound is greater than one, the current sink set is flattened to 2D ($z$-dimension is ignored) and partitioned geometrically by a horizontal or vertical line. Since each subset contains sinks from both dies, we potentially need many TSVs to connect them.

At the end of partitioning, we decide the TSV bound for each subset as follows: (1) estimate the number of TSVs required by each set, and (2) divide the given bound according to the ratio of estimated TSVs. The cut direction is determined such that the TSV bound is balanced in both subsets.

During the embedding and buffering step, the internal nodes of the 3D abstract tree are placed, and buffers are inserted under zero-skew constraint. The classic DME algorithm [6] is extended to generate topology embedding for the given 3D abstract tree. A cost function that considers capacitance of buffers, TSVs and wires is used in buffer insertion.

## 2.3 TSV-Buffer Insertion

Pre-bond testability of die-0 requires a fully connected clock tree on die-0 so that the minimum-skew clock signal is delivered to all FFs on die-0 using one test probe. As mentioned earlier, if multiple TSVs are used, the 3D tree construction step gives a 3D tree, where die-0 contains a fully connected tree and die-1 contains a forest. During pre-bond testing, we separate the two dies and test them individually. In this case, the 2D tree on die-0 can be used without any additional modification. However, the clock skew of this tree may no longer be zero because the downstream capacitances of some branches on die-1 are not present after the separation. This additional skew will slow down the testing process.

Our strategy to avoid this skew degradation during the pre-bond testing of die-0 is to employ so-called "TSV-buffers". A TSV-buffer is simply a buffer inserted right before a TSV. In our testing-aware DME (= TaDME) algorithm, we add a TSV-buffer for each inter-die connection that requires a TSV and route the tree accordingly under the zero-skew goal. In this case, the TSV-buffers are inserted on die-0, where the clock source is located. Since the buffers shield off all the downstream capacitance, die separation for pre-bond testing will not cause any change to the delay at the sinks on die-0. The outcome of TaDME is a zero-skew 3D tree that contains a zero-skew 2D tree on die-0 after die separation.

A key step in our TaDME algorithm is bottom-up recursive tree merging. Given a pair of zero-skew sub-trees to be merged, our goal is to locate the merging point and connect it to the root nodes of the sub-trees so that zero skew is maintained in the merged tree. Figure 1(a) shows the traditional merging process used in the original DME algorithm, where the location of a merging point $E$ is determined based on the parasitics of TSV, wires, the downstream capacitance and internal delay of the two sub-trees. In this case, if the right branch of the overall tree is removed, i.e., TSV, edge $(E, A)$, and $CT_2$, the delay from $E$ to $B$ will change due to the change on the downstream capacitance at point $E$. However, if merging with a TSV-buffer, the delay from $E'$ to $B$ in Figure 1(b) will not change even if we remove the right branch. This is because the TSV-buffer hides the downstream capacitance at point $E'$.

The following notations are used in Figure 1: $r$ and $c$ denote the unit length wire resistance and capacitance, respectively. $R_d$ is the output resistance of a buffer, $C_L$ is the input capacitance of a buffer, and $t_d$ is the intrinsic delay of a buffer. $R_{TSV}$ and $C_{TSV}$ are the resistance and capacitance of a TSV. Die-0 contains a subtree $CT_1$ with the root $B$. It has loading capacitance $C_{LB}$, and the internal delay from $B$ to the sinks of $CT_1$ is $t_B$. Similar symbols are used for $CT_2$. A clock wire of length $l$ is modeled as a $\pi$-type circuit with a resistor ($rl$) and two capacitors ($cl/2$). TSV is modeled as a wire with resistance of $R_{TSV}$ and capacitances of two $C_{TSV}/2$. Note that the downstream capacitance at the merging point $E'$ in Figure 1(b) is $cl_{E'B}+C_{LB}+C_L$ before *and* after the die separation for testing. Thus, TSV-buffers allow us to build a zero-skew 3D tree that contains a zero-skew 2D tree on die-0 after die separation.

During the bottom-up merging process, we require that the delay from $E'$ to sinks of $CT_1$ through $B$ (= $d_{E',CT_1}$) be equal to that from $E'$ to sinks of $CT_2$ through $A$ (= $d_{E',CT_2}$). That is,

$$d_{E',CT_1} = d_{E',CT_2}$$

where

$$\begin{aligned} d_{E',CT_1} &= rl_{E'B}(cl_{E'B}/2 + C_{LB}) + t_B \\ d_{E',CT_2} &= t_d + R_d(C_{TSV} + cl_{E'A} + C_{LA}) + \\ &\quad R_{TSV}(C_{TSV}/2 + cl_{E'A} + C_{LA}) + \\ &\quad rl_{E'A}(cl_{E'A}/2 + C_{LA}) + t_A \end{aligned}$$

and

$$l_{E'B} + l_{E'A} = L$$

where $t_A$ is the internal delay from $A$ to sinks of $CT_2$, and $C_{LA}$ is

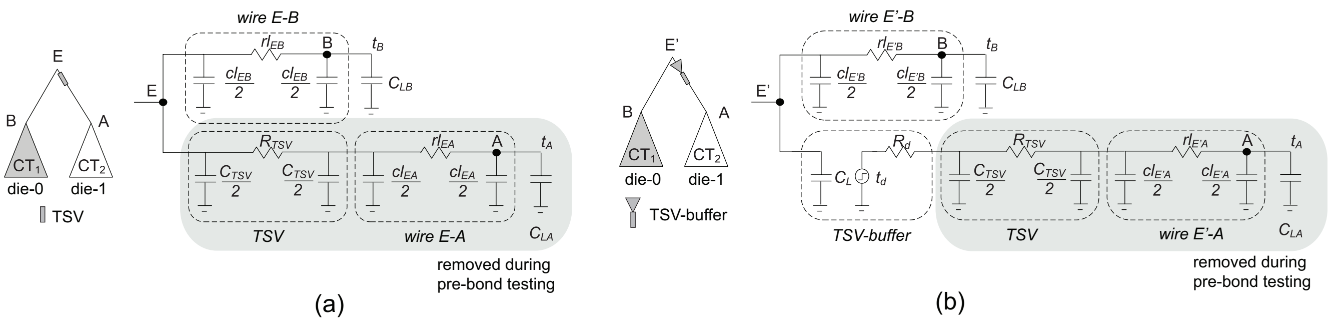

**Figure 1: (a) merging with a TSV, (b) merging with a TSV-buffer, where the separation of die-0 and die-1 does not affect the skew.**

the downstream capacitance of node $A$. $L$ is the merging distance between $A$ and $B$. The location of merging point, i.e., $l_{E'A}$ and $l_{E'B}$ can be determined by solving these equations.

## 2.4 Redundant Tree Insertion

Pre-bond testability of die-1 requires a fully connected clock tree so that the minimum-skew clock signal is delivered to all the FFs on die-1 using one test probe. As mentioned earlier, when multiple TSVs are used for wirelength reduction, the 3D tree construction step generates a 3D tree, where die-1 contains a forest. Therefore, our goal is to combine these sub-trees on die-1 into a single fully connected clock tree so that the clock skew is zero and the overall wirelength is minimized. We accomplish this by adding an additional tree, so called “redundant tree”, that connects the roots of the sub-trees while maintaining zero skew. We use this fully connected tree during the pre-bond testing of die-1. Note that the redundant tree is not to be used during the post-bond testing and operations. Our strategy is to use TGs (= transmission gates) to connect and disconnect the redundant tree.

The redundant tree routing is done using a conventional approach. Given the roots of the sub-trees on die-1, we construct a binary abstract tree in a top-down fashion. We then insert a TG at each root node. Next, we embed and buffer the abstract tree using the classical DME algorithm [6] under the zero-skew and minimal wirelength goals. Lastly, we connect the enable input of the TGs using an extra control wire. In order to minimize the routing resource overhead, we minimize the total wirelength of this control signal. We use the RMST-pack [11] for this purpose. Section 4.3 provides results on how significant this overhead (= redundant tree and TG control signal) is.

## 2.5 Putting It Together

Upon the completion of our algorithm, we obtain a fully connected zero-skew 2D clock tree for die-0 and die-1 each as well as a fully connected zero-skew 3D tree for the entire stack. In the case of die-1, we turn on the TGs to connect the redundant tree to the sub-trees on die-1. These two zero-skew trees are used during pre-bond testing. Once the pre-bond testing is completed, we turn off the TGs to disconnect the redundant tree from die-1. At this point, the original zero-skew 3D tree is used for post-bond testing and normal operations. We show in our experimental section that our approach that relies on the usage of multiple TSVs, TSV-buffers, TGs, and the control signal consumes significantly less power compared with a simple solution, where a single TSV is used to connect two separate zero-skew trees on die-0 and die-1.

## 2.6 Multiple-Die Extension

Our pre-bond testable clock tree algorithm for two-die stack can be easily extended to handle more than two dies. Our basic 3D tree construction algorithm presented in Section 2.2 generates a 3D tree, where the die that contains the clock source (= die-0 in this case) has a single fully connected tree, while all the other dies have a forest. The basic approach remains the same: during the bottom-up merging process, we insert a TSV-buffer at each TSV location on die-0 only. If the TSV does not connect to die-0, no TSV-buffer is required. Note that the TSVs connecting non-adjacent dies, e.g., die-0 and die-2, are assumed to be stacked on top of each other. In this case, we just need a single TSV-buffer on die-0. Once the TSV-buffer insertion and embedding/buffering are completed, we add redundant trees for all the other dies that contain a forest. We add transmission gates at the roots of all the sub-trees, and provide a global control signal to connect all the transmission gates on each die. The outcome of the whole process is: (1) a single zero-skew 3D clock tree for post-bond testing and normal operations, (2) a zero-skew 2D clock tree for each die to enable pre-bond testing.

# 3. SLEW-AWARE BUFFERING

## 3.1 Wirelength Balancing with Clock Buffers

Our 3D clock tree algorithm inserts two kinds of buffers: clock buffers and TSV-buffers. Clock buffers, as discussed in Section 2.2, are mainly used to control delay and skew. These clock buffers are usually inserted closer to the clock source and drive large loads to reduce the delay along the clock paths. The TSV-buffers, as discussed in Section 2.3, are inserted at every TSV location on the clock source die to make sure that the clock tree included on the clock source die maintains zero skew during pre-bond testing. Our observation indicates, however, that TSV-buffers usually cause the wirelength to be unbalanced during the bottom-up merging process. The reason is that if two sub-trees $CT_1$ on die-0 and $CT_2$ on die-1 are merged, we are forced to add a TSV-buffer on die-0. As shown in Figure 1(b), TSV-buffer insertion increases the delay from $E'$ to $CT_2$. Depending on the internal delays and downstream capacitance of the two sub-trees, if $t_A > t_B$, the TSV-buffer enlarges the unbalance of internal delays, causing $(E', B)$ to become longer. If the delay difference is very large, we may even require wire snaking to balance the delay for zero skew. This wirelength unbalance may cause the overall clock wirelength overhead on die-0 to become non-negligible.

Our strategy to tackle this problem is by adding extra clock buffers to balance the internal delays and thus the length of the related wires. Specifically, if the wirelength unbalance caused by TSV-buffer insertion is significant, we insert an extra clock buffer on the other branch to balance the internal delays. In Figure 1(b), we will add an extra clock buffer along $E'$-$B$. We observe that this delay balancing with extra clock buffer insertion eventually reduces the

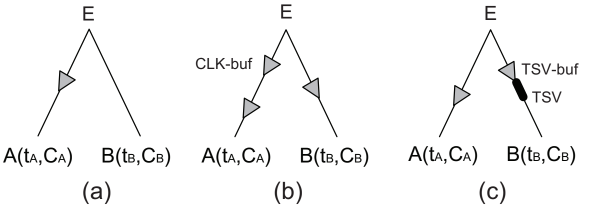

**Figure 2: Examples of clock/TSV-buffer insertion. (a) a clock buffer is inserted to balance the delay on both branches, where $t_A < t_B$, (b) multiple clock buffers are inserted if the wires are long and/or download capacitance is large, (c) a TSV-buffer is inserted along with a clock buffer to balance the delay.**

overall wirelength on die-0. We observe that the number of clock buffers used for wirelength balancing is usually low because the wirelength unbalance does not occur frequently.

## 3.2 Slew Rate Control with Clock Buffers

Clock slew rate control is an important reliability issue for high-speed clocking. If the slew rate is too low, i.e., if it takes too long time for the clock signal to rise to 1 or fall to 0, the FF setup and hold time are affected, which will eventually slow down the clock. Existing work on slew-aware clock tree synthesis relies on buffer insertion [12, 4, 5, 7]. Buffers are added along the clock paths so that the output load of each buffer is limited to a certain upper bound. This bound, denoted CMAX in the literature, is shown to be effective in improving the slew rate: smaller CMAX value improves the slew rate at the cost of more buffers inserted. Most existing works insert buffers to a given clock tree as a post process to improve the slew rate under various constraints including buffer area, clock power, etc. A limitation of this post-synthesis slew-aware buffering is that buffer insertion needs to be done carefully not to increase the clock skew. This may impose constraint on the location of buffers.

Our strategy to tackle the slew rate issue is by adding buffers under the CMAX constraint *during* clock tree synthesis. Specifically, we insert clock buffers, together with TSV-buffers, during the bottom-up merging process so that the CMAX constraint is satisfied for both types of buffers. We add clock buffers along the paths from the merging point to the sub-tree root nodes if the downstream capacitance at the merging point exceeds CMAX. Depending on the load, we may insert more than one clock buffer to meet the CMAX requirement.

Figure 2 shows several possible scenarios for clock/TSV-buffer insertion. In summary, our clock tree synthesis algorithm uses three criteria to insert buffers during the bottom-up merging process: (1) we add a TSV-buffer for every TSV connecting to the clock source die (for pre-bond testability), (2) we add a clock buffer if the wirelength between the merging point and two sub-tree roots are not balanced (for wirelength reduction), (3) we add clock buffers if the downstream capacitance of any buffer exceeds the given upper bound, namely CMAX (for slew rate control)

# 4. EXPERIMENTAL RESULTS

We implemented our algorithm using C++/STL on Linux. We use five benchmarks from the IBM suite [1] and four from the ISPD clock network synthesis contest suite [2]. Since these designs are for 2D ICs, we obtain 3D designs by randomly partitioning the clock sinks across multiple dies and scaling the footprint area by $\sqrt{2}$ and $\sqrt{4}$ for two-die and four-die stacks, respectively.

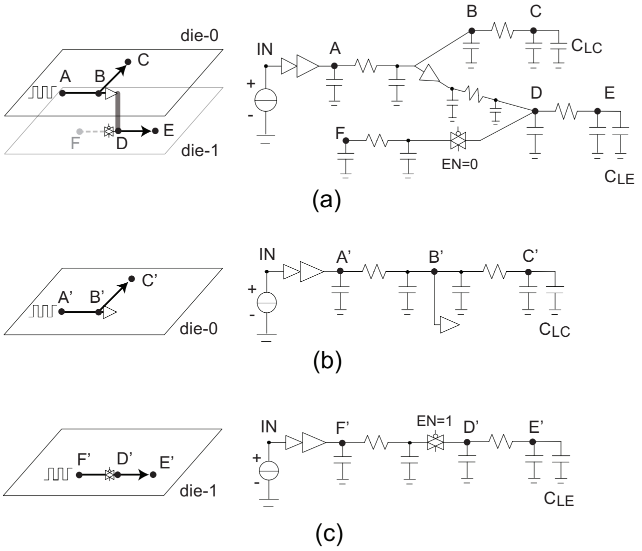

**Figure 3: Circuit models for (a) the post-bond 3D clock tree, (b) the pre-bond testable 2D clock tree on die-0, (c) the pre-bond testable 2D clock tree on die-1**

**Table 3: Impact of CMAX ($fF$) on skew ($ps$) and slew ($ps$) based on four-die stack of $r_1$. We compare the single-TSV and the multi-TSV approaches.**

| | Skew | | Max rise-slew | | Max fall-slew | |
|---|---|---|---|---|---|---|
| CMAX | Single | Multi | Single | Multi | Single | Multi |
| 150 | 22.6 | 5.6 | 37.1 | 37.4 | 32.8 | 33.0 |
| 175 | 22.0 | 6.3 | 43.9 | 44.0 | 38.7 | 38.6 |
| 200 | 8.8 | 6.7 | 51.5 | 50.5 | 45.5 | 44.3 |
| 225 | 11.3 | 7.3 | 58.7 | 54.0 | 52.4 | 47.4 |
| 250 | 9.7 | 8.3 | 67.4 | 59.7 | 60.1 | 52.4 |
| 275 | 12.4 | 11.4 | 76.4 | 71.0 | 68.5 | 62.5 |
| 300 | 10.5 | 13.3 | 86.6 | 80.8 | 78.2 | 71.5 |

We use the technology parameters based on 45nm PTM [3]: the unit-length wire resistance is $0.1\Omega/\mu m$, and the unit-length wire capacitance is $0.2fF/\mu m$. The sink capacitance values range from $5fF$ to $80fF$. The buffer parameters are: $R_d = 122\Omega$, $C_L = 24fF$, and $t_d = 17ps$. We use $10\mu m \times 10\mu m$ via-last TSVs with thinned die height of $20\mu m$. The TSV parasitics are: $R_{TSV} = 0.035\Omega$, and $C_{TSV} = 15.48fF$. Clock frequency is set to $1GHz$ with supply voltage of $1.2V$. Clock skew is constrained to 3% of the clock period. Clock slew constraint is set to 10% of the clock period. Correspondingly, the maximum load capacitance of each buffer (= CMAX) is $300fF$ for the slew rate control.

Out pre-bond testable clock routing algorithm generate zero-skew clock trees under the Elmore delay model. We will report all the clock-related metrics such as delay, skew, slew and power consumption based on SPICE simulation.

## 4.1 TSV-buffer and TG Model Validation

In pre-bond testable clock routing, we utilize two new circuit elements, namely TSV-buffer and transmission gates (= TGs), to facilitate pre-bond testing and post-bond testing/operations. TSV-buffers are used to shield off the downstream capacitance of die-0 (= the die that contains the clock source), which helps maintain zero skew during the pre-bond testing of die-0. TGs are inserted to support the pre-bond testing of other dies that contain sub-trees.

Figure 3 shows the equivalent circuit models used for the SPICE validation of TSV-buffers and TGs. We simulate a post-bond 3D

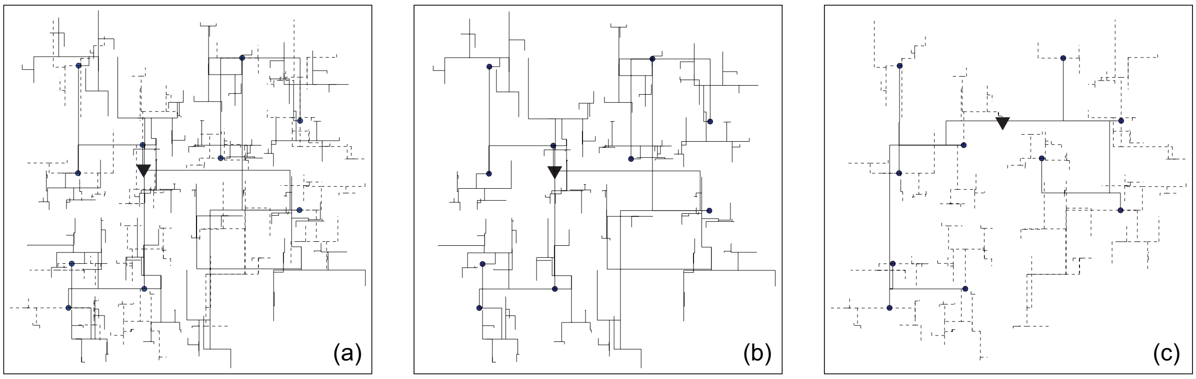

**Figure 4: Pre-bond testable clock trees of $r1$ in two-die stack under TSV bound of 10. (a) The post-bond 3D clock tree, where the solid and dotted lines denote the trees in die-0 and die-1, respectively, (b) the pre-bond testable 2D clock tree for die-0, (c) the pre-bond testable 2D clock tree for die-1, where the solid line is the redundant tree.**

**Table 1: Wirelength $(\mu m)$, power $(mW)$, and skew $(ps)$ results for the post-bond 3D clock tree, the pre-bond testable 2D clock tree for die-0 and die-1.**

| | | | post-bond 3D | | | pre-bond testable die-0 | | | pre-bond testable die-1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ckt | #Sinks | #TSVs | WL | Power | Skew | WL | Power | Skew | WL | WL-sub | WL-red | WL-TG | Power | Skew |
| $r_1$ | 267 | 57 | 227141 | 128.4 | 13.7 | 166691 | 103.0 | 13.5 | 150219 | 60450 | 89769 | 62732 | 68.2 | 13.0 |
| $r_2$ | 598 | 95 | 488987 | 274.1 | 14.2 | 328914 | 196.0 | 14.1 | 302023 | 160073 | 141950 | 109031 | 148.6 | 11.8 |
| $r_3$ | 862 | 183 | 616077 | 361.6 | 15.5 | 444156 | 280.5 | 15.5 | 429950 | 171921 | 258029 | 161561 | 201.9 | 16.2 |
| $r_4$ | 1903 | 265 | 1311290 | 763.2 | 15.5 | 889460 | 536.4 | 14.9 | 846980 | 421830 | 425151 | 259442 | 422.1 | 15.1 |
| $r_5$ | 3101 | 269 | 1998950 | 1115.0 | 29.1 | 1255760 | 715.9 | 29.1 | 1236417 | 743190 | 493227 | 310855 | 615.9 | 20.9 |
| $ispd09f11$ | 121 | 44 | 129391 | 73.3 | 9.4 | 99393 | 64.1 | 9.2 | 99169 | 29998 | 69171 | 51214 | 44.3 | 6.3 |
| $ispd09f12$ | 117 | 36 | 127763 | 71.2 | 6.8 | 96093 | 60.4 | 6.2 | 93625 | 31669 | 61956 | 42134 | 42.0 | 5.7 |
| $ispd09f21$ | 117 | 42 | 136676 | 75.6 | 5.0 | 107834 | 67.0 | 4.7 | 101968 | 28841 | 73127 | 52241 | 45.0 | 7.3 |
| $ispd09f22$ | 91 | 30 | 80977 | 46.8 | 15.3 | 61504 | 40.4 | 15.2 | 59870 | 19473 | 40397 | 29449 | 26.4 | 14.9 |
| RATIO | | | 1.00 | 1.00 | 1.00 | 0.72 | 0.79 | 0.97 | 0.69 | 0.28 | 0.41 | 0.29 | 0.57 | 0.94 |

clock tree for two-die stack along with the pre-bond testable 2D clock tree on die-0 and die-1. Node $A$ is the clock source for post-bond operation. Sink $C$ on die-0 and sink $E$ on die-1 have loading capacitances of $C_{LC}$ and $C_{LE}$, respectively. Node $B$ and $D$ are connected by a TSV-buffer and its TSV. The edge $(D, E)$ is a sub-tree on die-1 and is connected to $F$, its pre-bond testing clock source, via a TG. $C_{LC}$ and $C_{LE}$ are set to $5fF$. Wire $(A, B)$, $(B, C)$, $(D, E)$ and $(F, D)$ all have length $500\mu m$.

First, we observe from SPICE simulation that the delay from $A$ to $C$ in Figure 3(a) is $42.21ps$, which is the same as that from $A'$ to $C'$ in Figure 3(b). This verifies that our TSV-buffer maintains the delay to the sinks on die-0 after separating die-1 for pre-bond testing. Second, the TG has $14.2fF$ capacitance between node $D$ and the ground when it is off. This TG completely blocks the clock signal from $A$ to $F$. When TG is on for the pre-bond testing of die-1, however, it has $108\Omega$ between its input and output nodes, $16.4fF$ between its input and the ground, and $18.4fF$ between its output and the ground. The intrinsic delay of a TG is $1.04ps$. Under this model, the calculated delay from $F'$ to $E'$ is $54.13ps$, which closely matches with the simulated delay of $54.14ps$.

## 4.2 Sample Trees and Waveforms

Figure 4 shows a series of pre-bond testable clock trees of the circuit $r_1$ from the IBM suite based on TSV upper bound of 10. Figure 4(a) is the zero-skew 3D clock tree for post-bond testing and normal operations. The solid and dotted lines represent the clock trees on die-0 and die-1, respectively. It contains 10 TSVs denoted by the black dots. The triangle is the clock source. Note that die-1 contains many sub-trees (= dotted lines) that are not connected. Figure 4(b) shows the zero-skew pre-bond testable 2D clock tree for die-0, which is identical to the solid clock tree in Figure 4(a). Figure 4(c) shows the zero-skew pre-bond testable 2D clock tree for die-1, which contains a redundant tree that connects the root nodes of the sub-trees on die-1 (= dotted lines).

Figure 5 shows two groups of clock waveforms for benchmark $r_5$, where each group contains 25 waveforms for 25 sinks in each tree. The first group (shown on top) is from the post-bond 3D clock tree, whereas the second group (shown on bottom) is from the pre-bond testable 2D clock tree for die-0. We first observe that the waveforms among 25 sinks are almost identical, which is desirable. In addition, the two groups have similar waveforms, which demonstrates that the TSV-buffer helps maintain waveforms for the post-bond and pre-bond testing. Second, the SPICE simulation shows that the clock skew among all sinks in both cases is $29.1ps$, which can also be observed by the width of waveforms at 50% Vdd. Third, the maximum slew rate is $88.4ps$, which is measured as the rise time from 10%-to-90% of Vdd (or 90%-to-10%). Both the skew and slew values satisfy our constraints (3% and 10% of clock period, respectively).

## 4.3 Wirelength, Skew, and Power Results

Table 1 shows the wirelength $(\mu m)$, power consumption $(mW)$, and skew $(ps)$ results for the post-bond 3D clock tree (= denoted $T_{3D}$), the pre-bond testable 2D clock tree for die-0 (= denoted $T_0$) and die-1 (= denoted $T_1$). For die-1, we report the total wirelength (= WL), wirelength of the sub-trees (= WL-sub), redundant tree (= WL-red), and the TG control signal (= WL-TG). In this case, the WL of $T_1$ is equal to the sum of WL-sub and WL-red. In addition, the WL of $T_{3D}$ is sum of the WL of $T_0$ and the WL-sub of $T_1$.

Based on the wirelength-related columns, we see that (1) the total

Table 2: Comparisons of wirelength ($\mu m$), power ($mW$), and skew ($ps$) between the single-TSV and the multi-TSV approaches.

| | | | Single TSV | | | | Multi-TSV | | | | | reduction % | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ckt | #Sinks | #Bufs | WL | Power | Skew | #TSVs | #Bufs | WL | Power | Skew | WL | Power |
| Two-die | $r_1$ | 267 | 327 | 279796 | 145.0 | 12.7 | 57 | 324 | 227141 | 128.4 | 13.7 | 18.8 | 11.4 |
| | $r_2$ | 598 | 693 | 600880 | 310.6 | 12.5 | 95 | 684 | 488987 | 274.1 | 14.2 | 18.6 | 11.8 |
| | $r_3$ | 862 | 928 | 765397 | 404.3 | 16.1 | 183 | 925 | 616077 | 361.6 | 15.5 | 19.5 | 10.6 |
| | $r_4$ | 1903 | 1982 | 1576510 | 848.7 | 15.3 | 265 | 1963 | 1311290 | 763.2 | 15.5 | 16.8 | 10.1 |
| | $r_5$ | 3101 | 2528 | 2344960 | 1242.0 | 22.2 | 269 | 2449 | 1998950 | 1115.0 | 29.1 | 14.8 | 10.2 |
| | $ispd09f11$ | 121 | 212 | 168500 | 85.4 | 7.6 | 44 | 201 | 129391 | 73.3 | 9.4 | 23.2 | 14.1 |
| | $ispd09f12$ | 117 | 215 | 164966 | 84.2 | 5.8 | 36 | 193 | 127763 | 71.2 | 6.8 | 22.6 | 15.5 |
| | $ispd09f21$ | 117 | 226 | 180867 | 89.9 | 9.4 | 42 | 211 | 136676 | 75.6 | 5.0 | 24.4 | 15.9 |
| | $ispd09f22$ | 91 | 106 | 106401 | 53.2 | 15.1 | 30 | 111 | 80977 | 46.8 | 15.3 | 23.9 | 12.1 |
| Four-die | $r_1$ | 267 | 318 | 272355 | 141.8 | 10.5 | 325 | 248 | 160394 | 111.4 | 13.3 | 41.1 | 21.4 |
| | $r_2$ | 598 | 700 | 582115 | 304.5 | 14.4 | 434 | 647 | 353646 | 233.9 | 15.7 | 39.2 | 23.2 |
| | $r_3$ | 862 | 945 | 735299 | 398.0 | 14.9 | 718 | 922 | 442903 | 317.1 | 13.7 | 39.8 | 20.3 |
| | $r_4$ | 1903 | 1956 | 1532220 | 831.1 | 14.8 | 1651 | 2011 | 908375 | 675.6 | 16.5 | 40.7 | 18.7 |
| | $r_5$ | 3101 | 2939 | 2312930 | 1272.0 | 22.2 | 2469 | 3134 | 1368370 | 1041.0 | 20.3 | 40.8 | 18.2 |
| | $ispd09f11$ | 121 | 216 | 159752 | 83.1 | 8.4 | 129 | 176 | 93440 | 60.0 | 5.8 | 41.5 | 27.8 |
| | $ispd09f12$ | 117 | 208 | 155542 | 80.9 | 8.9 | 114 | 160 | 90281 | 56.8 | 10.2 | 42.0 | 29.7 |
| | $ispd09f21$ | 117 | 212 | 163816 | 83.0 | 17.8 | 102 | 160 | 99179 | 58.4 | 7.8 | 39.5 | 29.6 |
| | $ispd09f22$ | 91 | 99 | 98123 | 48.7 | 18.0 | 81 | 88 | 57342 | 36.1 | 14.7 | 41.6 | 25.9 |

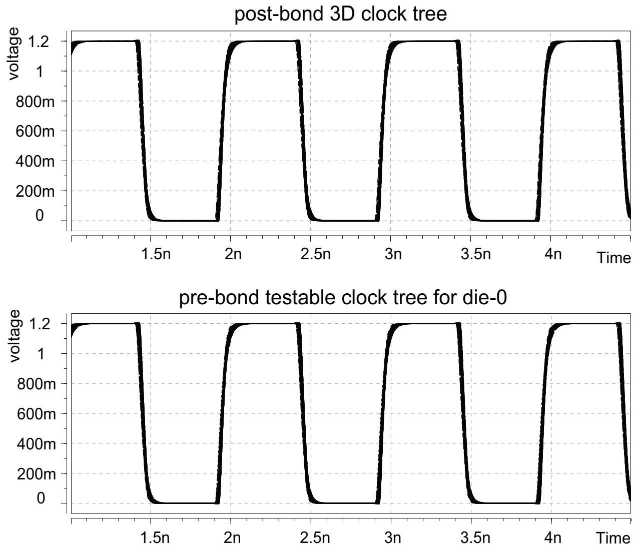

**Figure 5: Clock waveforms of the post-bond 3D clock tree and the pre-bond testable 2D clock tree for die-0. We superimpose the waveforms of 25 clock sinks in $r_5$. Clock frequency is $1GHz$, skew is $29.1ps$, and maximum slew rate is $88.4ps$.**

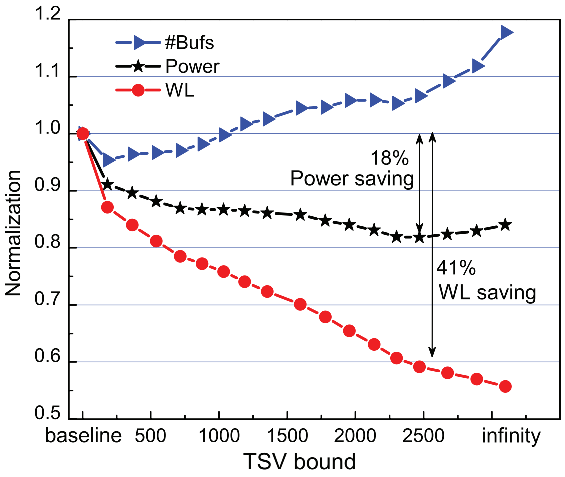

**Figure 6: Impact of TSV bound on wirelength, buffer count, and power consumption ($mW$) based on four-die stack of $r_5$. The baseline is the single-TSV approach.**

wirelength (= WL) of $T_0$ and $T_1$ are comparable, (2) in several cases, the wirelength of the redundant tree (= WL-red) is about 2x the total wirelength of the sub-trees (= WL-sub) on die-1, (3) in several cases, the wirelength of the TG control signal (= WL-TG) is about half the redundant tree in die-1 (= WL-red). Note that the redundant tree and the TG control signal are used only during the pre-bond testing of die-1. This non-negligible overhead is compensated by the significant power saving to be discussed in Section 4.4. The skew values do not exceed $30ps$, which satisfies our skew constraint that is set to 3% of the clock period. In terms of clock power consumption, die-0 consumes more power than die-1 primarily due to the TSV-buffers added on die-0.

## 4.4 Comparison with Single-TSV Approach

Our baseline 3D clock tree contains a single fully connected zero-skew clock tree on each die, and these trees are connected with a single TSV for two-die stack and a single "stacked TSV" for more than two dies in the stack. Table 2 shows the comparisons of wirelength, clock power and skew results based on SPICE simulation. The runtime for each circuit is less than one second in all cases.

First, our multi-TSV approach significantly outperforms the single-TSV approach in terms of wirelength: 14.8%-24.4% for two-die stack, and 39.2%-42.0% for four-die stack. Similarly, power saving for the clock tree is 10.1%-15.9% for two-die, and 18.2%-29.7% for four-die stack. These results convincingly demonstrate the benefits of our multi-TSV approach. Second, the #Bufs columns show the total number of buffers used in the trees, including both the clock buffers and TSV-buffers. We see that the total number of buffers used is comparable between the single-TSV and the multi-TSV approaches. In the case of the single-TSV approach, buffers are inserted mainly to control wirelength and slew in each die. As for our multi-TSV approach, it shows different trends in terms of buffer usage between the two-die and four-die cases: most of the buffers are clock buffers that are used for wirelength and slew

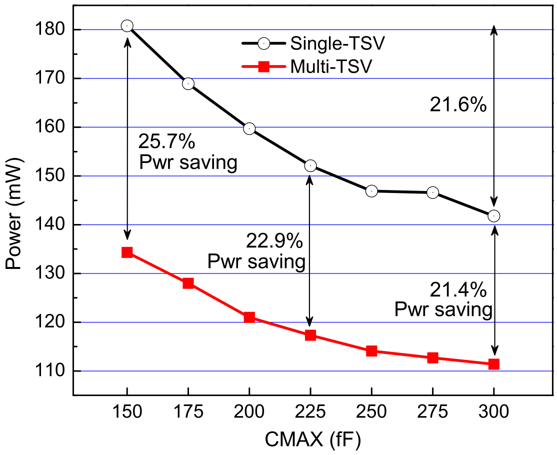

**Figure 7: Impact of CMAX ($fF$) on power consumption ($mW$) based on four-die stack of $r_1$.**

control in two-die stack; in contrast, most of the buffers are TSV-buffers for the four-die cases. Since the TSV-buffers also have positive impact on wirelength and slew in the four-die cases, we do not need too many clock buffers. Third, the clock skew values are all below the constraint of $30ps$ in both cases.

## 4.5 Wirelength and Power Results

Figure 6 shows the impact of TSV bound on wirelength, buffer count, and clock power consumption. These metrics are normalized to the baseline results, which are based on the single-TSV approach. The x-axis corresponds to the TSV bound used to build our multi-TSV pre-bond testable 3D clock tree. Note that the actual TSV usage may be different from this bound because the clock tree synthesis algorithm determines itself the optimal number of TSVs to be used for wirelength minimization. When TSV bound is set to infinity, the actual TSV usages is 3097 for benchmark $r_5$.

We first observe that the wirelength consistently reduces as more and more TSVs are used in our 3D pre-bond testable clock tree. The wirelength saving reaches 45% if the TSV bound is set to infinity. This shows that TSVs in general help reduce the overall wirelength of 3D clock tree. Second, the total number of buffers used (counting both the clock buffers and TSV-buffers) increase as more and more TSVs are used. This is mainly due to the TSV-buffers inserted for pre-bond testability. Taking both the wirelength and buffer trends into consideration, the power consumption reduces, consistently but slowly, as more and more TSVs are used. The maximum power saving for $r_5$ is around 18% compared with the single-TSV case, when the 3D clock tree uses around 2500 TSVs across all four dies. If more than 2500 TSVs are used, the power consumption goes back up, mainly due to the excessive TSV-buffer insertion. This trend allows us to choose the right TSV bound for a given power budget. If the power saving of 10% is required, the TSV bound is set to 300.

## 4.6 Impact of CMAX on Power and Slew

Table 3 shows the impact of CMAX (= the maximum output load each buffer can drive) on skew, maximum rise-slew and maximum fall-slew among all sinks on all dies. We use four-die stack of benchmark $r_1$ and compare the single-TSV with our multi-TSV approaches. We observe that as the CMAX value increases, the maximum rise and fall slews for both single-TSV and multi-TSV cases increase. In other words, tighter CMAX means better slew. All of the slew values are below the constraint, 10% of the clock period, which is $100ps$. The slew values are slightly smaller in case of multi-TSV, mainly due to (slightly) more buffers inserted for slew control. In terms of skew, the trend is not obvious for the single-TSV case. However, skew tends to reduce with a tighter CMAX value for the multi-TSV case. The main reason is that wirelength is shorter in these cases, which cause the clock buffers added for slew control to have positive impact on delay and skew as well.

Figure 7 shows the impact of CMAX on clock power consumption. We use four-die stack of $r_1$ for this experiment. The overall trend is the same in both single-TSV and multi-TSV cases: tighter CMAX results in more power consumption. This is because more clock buffers are inserted to meet the tight CMAX constraint. However, the power benefit of the multi-TSV case over the single-TSV case remains consistent regardless of the CMAX value.

# 5. CONCLUSIONS

In this paper, we studied how to construct a clock tree for 3D stacked ICs so that each individual die can be tested before bonding. Our solution utilizes multiple TSVs for wirelength and clock power saving, which in turn necessitate new circuit elements, namely, TSV-buffers, redundant tree, and transmission gates, to support low-skew and low-power testing and operations. We also studied the impact of buffer insertion on slew rate in 3D stacked ICs clocking. SPICE results show that our method based on multiple TSVs significantly reduces the wirelength and power of the 3D clock tree from a baseline approach that uses only one TSV.

# 6. REFERENCES

[1] GSRC Benchmark, http:// vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST.

[2] ISPD Contest 2009, http://www.sigda.org/ispd/contests/ispd09cts.html.

[3] Predictive Technology Model, http://www.eas.asu.edu/ ptm/.

[4] C. Albrecht, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky. On the Skew-Bounded Minimum-Buffer Routing Tree Problem. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(7):937–945, July 2003.

[5] C. J. Alpert, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky. Minimum Buffered Routing with Bounded Capacitive Load for Slew Rate and Reliability Control. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(3):241–253, March 2003.

[6] K. Boese and A. Kahng. Zero-Skew Clock Routing Trees With Minimum Wirelength. In *ASIC Conference and Exhibit, 1992., Proceedings of Fifth Annual IEEE International*, pages 17–21, 1992.

[7] S. Hu, C. Alpert, J. Hu, S. Karandikar, Z. Li, W. Shi, and C. Sze. Fast Algorithms for Slew-Constrained Minimum Cost Buffering. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 26(11):2009–2022, Nov. 2007.

[8] D. L. Lewis and H.-H. S. Lee. A Scan-Island Based Design Enabling Pre-bond Testbility in Die-Stcked Microprocessors. In *IEEE International Test Conference*, pages 1–8, 2007.

[9] D. L. Lewis and H.-H. S. Lee. Testing Circuit-Partitioned 3D IC Designs. In *Proc. Int. Symp. on VLSI*, 2009.

[10] J. Minz, X. Zhao, and S. K. Lim. Buffered Clock Tree Synthesis for 3D ICs Under Thermal Variations. In *Proc. Asia and South Pacific Design Automation Conf.*, pages 504–509, 2008.

[11] RMST-Pack. http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST/.

[12] G. E. Tellez and M. Sarrafzadeh. Minimal Buffer Insertion in Clock Trees with Skew and Slew Rate Constraints. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 16(4):333–342, April 1997.

[13] X. Wu, P. Falkenstern, and Y. Xie. Scan Chain Design for Three-dimensional Integrated Circuits (3D ICs). In *Proc. IEEE Int. Conf. on Computer Design*, pages 208–214, 2007.